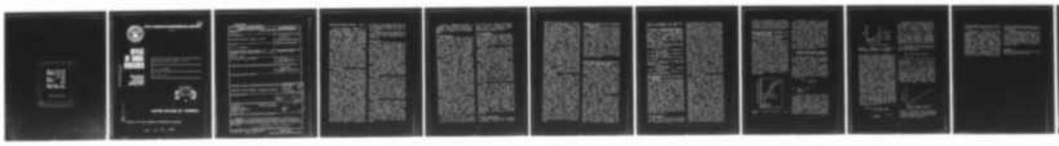
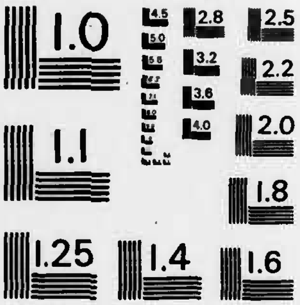
AD-A136 317 WORLD COMPUTER CONGRESS (9TH) IFIP (INTERNATIONAL 1/1
FEDERATION OF INFORMATION PROCESSING) 83(U) OFFICE OF
NAVAL RESEARCH LONDON (ENGLAND) J F BLACKBURN
UNCLASSIFIED 24 OCT 83 ONRL-C-17-83 F/G 9/2 NL



END
DATE
FILMED
1 84
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

12

ONR LONDON CONFERENCE REPORT

C-17-83

DEPARTMENT OF THE NAVY
O.N.R. LONDON BRANCH

OFFICE OF NAVAL RESEARCH

BRANCH
OFFICE
LONDON
ENGLAND

NINTH WORLD COMPUTER CONGRESS: IFIP 83

DR. J.F. BLACKBURN

24 October 1983

UNITED STATES OF AMERICA

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER C-17-83	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Ninth World Computer Congress: IFIP 83		5. TYPE OF REPORT & PERIOD COVERED Conference
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Dr. J. F. Blackburn		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS US Office of Naval Research Branch Office London Box 39 FPO NY 09510		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE 24 October 1983
		13. NUMBER OF PAGES 7
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computers
Processing units
Computer architecture
Computer logic
Computer programming

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Highlights of the conference included discussions of the design of Ada, the VLSI/ULSI chip, logic programming, multiprocessor data bases, high-performance computers, fifth-generation computers, dataflow machines, and high-speed processors.

ULTRALARGE SCALE INTEGRATION

VERY LARGE SCALE INTEGRATION

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

NINTH WORLD COMPUTER CONGRESS: IFIP 83

The International Federation of Information Processing (IFIP) is a multinational federation of professional and technical organizations concerned with information processing. The aims of IFIP are to promote information science and technology by fostering international cooperation; stimulating research, development, and applications; furthering the dissemination and exchange of information; and encouraging education in information processing.

The IFIP held its ninth congress in Paris from 19 through 23 September 1983. There were more than 2000 participants from Europe, North America, Africa, Asia, Japan, Australia, and New Zealand About 1500 of the participants were from Europe. *Information Processing 83*, the proceedings of the conference, is available from Elsevier Science Publishers B.V., P.O. Box 1991, 1000 BZ Amsterdam, The Netherlands.

The principal opening address was given by Pierre Mauroy, Prime Minister of France. His theme was the need to meet the present industrial challenge through the use of information and communications technology. He stressed the importance the French government has placed on electronics through establishing a "filiere electronique" (electronic sector), which promotes research and development in information science and telecommunications. A great deal of government research has been transferred to private industry. Considerable emphasis is being given to computer-assisted instruction in education and to computer-assisted development in science and engineering. Mauroy made a strong statement on the need for international cooperation, which is fostered by IFIP. He also spoke of the increasingly important role robots are playing in world industry. This led him to emphasize the need for training young workers in electronics to better prepare them for the opportunities that will exist in a society with increasing use of information systems snd robots. Finally, he mentioned the interest of the government of France in encouraging advancement in the developing countries and the vital role information science will play in that effort.

Following the opening speeches, the congress was divided into parallel sessions. This report highlights some of the papers.

Design of Ada

Jean D. Ichbiah (France) gave an invited paper titled "On the Design of Ada." The Ada programming language was designed in accordance with a set of specifications issued by the US Department of Defense. The domain of application defined by these specifications is that of "embedded computer systems." Because of the requirements of such systems, the specifications put great emphasis on language features that encourage reliability and maintainability.

Another significant aspect of Ada is that its design takes account of the likelihood that software development will be increasingly distributed and that software systems will be assembled from independently produced software components.

Several features of the Ada language are directly related to the feasibility of the distributed production of software components. The first and most important of these is the package feature, the main form of module in Ada. A central idea in the design of packages is the textual separation of the user interface from the implementation. The user interface, called the package specification, is actually what would appear in a catalogue of software components.

For several reasons, packages may be expected to appear in compiled form rather than in source form in a library of software components. Hence Ada packages can be compiled separately.

A programmer using several independently developed packages for a given application has little control over the choices of names made by the producers of these packages. This fact has affected the naming rules of the language.

Finally, a parameterized form of a package, called a generic package, is provided. Parameters of generic packages can be values, types, and subprograms.

The VLSI/ULSI Chip

M. Ashikawa (Japan) served as chairman of a panel discussion dealing with the VLSI/ULSI chip. The first field of application of the microcomputer is what is called the single-chip microcomputer, used in the consumer electric field, automotive applications, household appliances, personal computers, and domestic video game computers. Multifunction, low power requirement, and low cost are important features.

The second field of application mainly takes the place of the minicomputer and the business computer and applies to industrial applications, measurement applications, data processing applications, the personal computer, and the office computer. High performance is a main objective; 16-bit and 32-bit microprocessors are used. By means of ultramodern fine patterning technology, the microprocessor in this field takes advantage of the high-level integration and the high speed of the metallic oxide semiconductor large scale integration (MOS/LSI). The quality of software is especially important in this field--as it is in that of the business computer. The main focus is on trends in operating systems and high-level language, and on the state of the art in very large scale integration/ultralarge scale integration (VLSI/ULSI).

The third field of application is signal processing. At present, the main application is in speech and sound processing, typically sound recognition. Once an analog signal is changed into a digital signal, digital signal processing (with its main function of high speed multiplication and addition operations) is performed. Growth in this field will lead to multiplexed analog and digital technology.

Expectations for the future include 1 million circuits per chip, implementation of high-level languages like Ada and Pascal, more powerful addressing modes, virtual addressing, larger address space, more pipelined systems, and greater microprogram control.

Logic Programming

An invited paper on logic programming was given by Robert Kowalski (UK). Logic programming has recently received considerable attention because the Japanese have chosen it as the core computer language for the Fifth Generation project.

Because Prolog, the computer language developed by Colmerauer and his colleagues in 1972 in Marseille, is an abbreviation of "Programming Logic," it has been confused with logic programming. Prolog is based upon logic programming in much the same way that LISP is based upon the Lambda calculus.

The notion of logic programming is itself ill-defined. In its most modest form, it refers only to programming with Horn clauses. (A Horn clause logic program consists of a set of rules and assertions in which each assertion expresses an atomic relationship among individuals, for example: John likes Mary. Each rule has the form: A if B and C and . . . and D of a single conclusion A and one or more conditions B, C, D.)

However, practical experience has shown that this modest notion is too restrictive. Horn clause programming had to be extended in various ways, as Kowalski noted. Logic programming was compared with rule-based artificial languages and functional programming. The main problems to be solved are: (1) the provision of features such as those offered by high-order functions in functional programming languages, and (2) the frame problem and its relationship with destructive assignment. A combination of object language and metalanguage provides a promising approach to the solution of these problems.

Functional Programming

Two papers were given on functional programming. Manfred Broy (Federal

Republic of Germany) spoke on "Applicative Real-Time Programming." In contrast to sequential programming, where time considerations are irrelevant for the functional behavior of programs and their correctness, in concurrent and communicating programs questions of relative speed and of time generally have decisive influences on the results. So the mathematical semantics of nonsequential programs are not independent of considerations of time and relative speed. Broy introduced an applicative programming language and used fixed-point theory to formally specify its semantics. An explicit discrete time is included in the semantic domain, such that the functions written in the applicative programming language map objects or streams of objects arriving at certain time points onto objects or streams of objects produced at certain time points. The language includes time delay functions as well as time comparison predicates. The relationship of the language to languages for writing concurrent programs was discussed, with time abstracted to causality flow and nondeterminism.

"Functional Programming With Streams" was the title of a paper given by Tetsuo Ida (Japan). The notion of a stream to encapsulate the (possibly) infinite flow of data has been used in several programming disciplines. The use of a stream makes a certain class of programs simpler and clearer. A method for processing a stream is to consider a stream as a recursive data structure and to evaluate it to the extent that only the necessary part of the data structure is constructed. Ida tackled streams from the aspect of iteration, in the context of functional programming. A stream is viewed as an infinite number of elements that are generated one in each iterative loop. The stream is formulated as a natural extension of iteration. Streams are embedded in a functional programming system, which permits clear and efficient treatment of streams. A set of primitive functions for stream programming was given, together with an outline of this implementation.

Design of a Multiprocessor Relational Data Base System

Georges Gardarin (France) described the Systeme d'Accis à des Bases Relationnelles (SABRE), which is a portable data-management system, mainly designed for a multi-microprocessor configuration. A first generation of the system is operational at Institut National de Recherche en Informatique et en Automatique (INRIA) on the Multics operating system. SABRE presents several original ideas. These include clustering of data, multiple views of a data base, algorithms for relational joins, concurrency control, integrity control, and query optimization. The objectives and the functional architecture of the SABRE system were described.

Architectural Concepts for High-Performance, General-Purpose Computers

Gene M. Amdahl (US) gave a retrospective of high-performance computers that he helped design. These computers include the WISC, IBM 704, IBM 709 and STRETCH, System/360 model 91/95, Amdahl 470 series, and the Trilogy computer.

The Amdahl 470 series is compatible with the IBM System/370 computers. The series was designed to optimize the cost performance at a moderately high performance level. The computers were unique, however, in bringing into being the world's first large-scale integrated circuits employing high-performance, emitter-coupled logic. This remained a first for almost the full market life of the series, some 6 years.

The project being carried out at Trilogy involves the development of an ultra-large-scale integrated-circuit chip. The level of integration encompasses a complex function on a single chip. The chips contain a high-performance, emitter-coupled-logic family. The computer being built will be System/370 compatible, and is planned to significantly outperform all contending compatibles. The computer will also place considerable emphasis on the scientific capabilities, providing very high general computing performance and vector processing. Announcement of specific performance and functional details will be made in late 1983.

Ease of Use: A System Design Challenge

Lewis M. Branscomb (US) gave an invited paper on high-performance computers. He made suggestions in four main areas: system architecture, software development tools, needed fundamental and applied research, and educational policy.

Some guidelines were given for improving system architecture to make the system easier to use:

1. User interface separated from the rest of the system
2. Layered interface adaptable to the expertise of the user
3. Media translatability for input and output
4. Facilities in the system for behavioral measurement of the user characteristics.
5. A synonym table
6. System adaptability to changing needs of the users.
7. Architectural adaptation to basic human capabilities.

The Architectures in the Fifth Generation Computers

This subject was discussed by Tohru Moto-Oka (Japan). The fifth generation computers are being developed predominantly for use with knowledge information processing systems, expected to come into widespread use in the 1990s. Inference machines and relational algebra machines are typical of the core processors which constitute the fifth generation computer systems. A new logic programming language is being designed for use with the fifth generation kernel language to act as the interface between hardware and software in these machines. Several examples of proposed architectures were described. VLSI-oriented architecture and parallel processing control mechanisms are the main research topics.

Dataflow Machines

Two papers were given on dataflow machines. First, John Gurd (UK) spoke on "Preliminary Evaluation of a Prototype Dataflow Computer." Despite widespread interest in dataflow computing techniques, relatively little is known about the performance of the few hardware systems that are now operational. Prototype hardware based on the tagged token model of dataflow computation has recently been constructed at the University of Manchester. The hardware structure is a pipelined ring which exhibits variable processing rate when the number of active homogeneous function units are adjusted--up to a limit determined by the pipeline beat period. The system is currently running small user programs at maximum rates between 1 and 1.8 million instructions per second (MIPS). It is exhibiting near-linear speedup with respect to the number of function units for a wide variety of programs which have sufficient parallelism to occupy all of the function units and pipeline stages. An immediate increase in performance, to 3 MIPS and with improved speedup characteristics, is possible for a single ring. In the long term, use of multiple-rings offers the possibility of incrementably expandable computing power.

Jack B. Dennis (US) spoke about "The MIT Dataflow Engineering Model." Dataflow computers differ from conventional computers in that new concepts of program structure, new programming languages, and new methodologies are needed for their effective use. At the Massachusetts Institute of Technology (MIT) laboratory of computer science, an engineering model has been constructed consisting of eight processing units coupled by a packet communication network built of two-by-two routers. This facility is being used to study issues and evaluate alternatives in the design of high-performance dataflow computing systems. Principal aspects being addressed are: (1) development of good machine code structures for computations basic to numerical physics problems, especially structures capable of handling massive data bases composed of large arrays of numerical data; (2) the design of instruction sets suitable for dataflow processing elements; (3) evaluation of packet-switched interconnection networks; (4) development of methodologies for designing the custom devices required in a prototype machine and for providing the level of fault tolerance to achieve reliable system operation.

How to Achieve High Performance for High-Speed Processors

This was a panel discussion session; the remarks of the chairman, B.L. Buzbee (US) are summarized here.

The growth rate in execution bandwidth of high-speed processors is diminishing. This is illustrated in Figure 1, which shows the execution bandwidth of some high-speed processors over the era of electronic computation. The asymptote to the curve is about 3 billion operations per second. We must ask whether the curve will accurately forecast performance in view of new developments in VLSI circuit technology. If the curve continues to forecast the future accurately, then we are left with the conclusion that a single processor has a maximum performance level and that state-of-the-art equipment is approaching it. An obvious way to circumvent this maximum is through the use of parallelism.

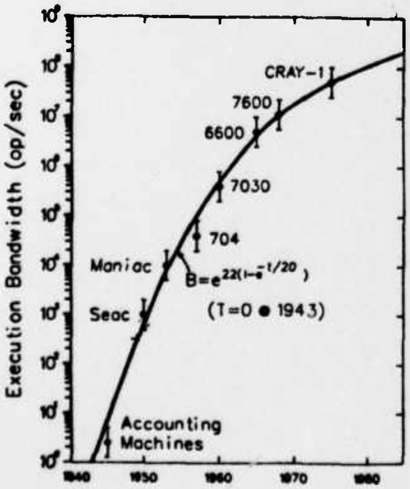


Figure 1. Trend in execution bandwidth.

Asynchronous systems of a few tightly coupled high-speed processors are a natural evolution from high-speed uniprocessor systems. Systems with two to four processors will soon be available: the Cray X-MP, the Cray 2, and the Control Data System 2XX. Systems with eight to 16 processors are likely by the early 1990s. What are the prospects of using the parallelism in such systems to achieve high speed in the execution of a single application?

The key issue in parallel processing a single application is speedup as a function of the number of processors used. Speedup is defined as

$$S_p = \frac{\text{execution time using one processor}}{\text{execution time using p processors}}$$

To estimate performance of a tightly coupled system on a single application, we use a model of parallel computation introduced by W. Ware (1983).

Let p equal the number of processors; α equals the percent of parallel processable work in the application. Assume at any instant that either all p processors are operating or only one processor is operating. If we normalize the execution time using one processor to unity, then

$$S_p = \frac{1}{(1-\alpha) + \frac{\alpha}{p}}$$

and $\left.\frac{dS_p}{d\alpha}\right|_{\alpha=1} = p^2-p.$

Figure 2 shows the Ware model of speedup as a function of α for a four-processor, an eight-processor, and a 16-processor system. There is a low speedup for α less than 0.9; therefore, to achieve significant speedup, we must have highly parallel algorithms. It is not evident that algorithms now used on uniprocessors contain the requisite parallelism. If not, research will be required to find suitable alternatives.

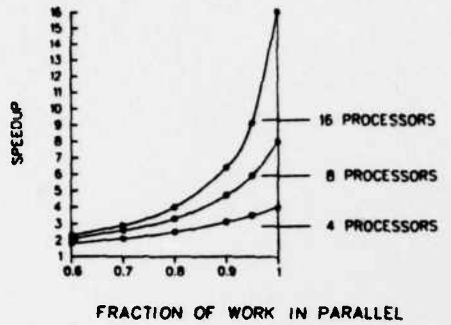


Figure 2. Speedup as a function of parallelism and number of processors.

When highly parallel algorithms are available, they must be implemented with care because α spans the entire application. Quadratic behavior of the derivative of α means that a small change in it produces a large change in speedup. Figure 2 also gives the performance of vector processors as a function of work vectorized, where p is the relative performance of the vector and scalar states. Ware's model is inadequate in that it assumes that the same instruction stream will be executed on a parallel system that is executed on a single processor. This is seldom the case because synchronization and communication in asynchronous systems usually require execution of instructions that are not present in a uniprocessor implementation. Also, parallel algorithms may require additional instructions. To correct this, a term, σ(p), is added to the execution time for parallel implementation; σ is non-negative and usually monotonically increases with p. It is a function not only of p, but also of the algorithm and the architecture and of α.

$$\text{Let } \bar{S}p = \frac{1}{(1-\alpha)+\frac{\alpha}{p}+\sigma\,(p)}$$

$$= \frac{p}{1+p\;\sigma(p)} \qquad \text{at } \alpha=1$$

Consequently, in general the maximum speedup of a real system will be less than p, and may be significantly less ($\bar{S}p < 1$ for small α). Further, $\bar{S}p$ will have a maximum for sufficiently large p; that is, $\frac{\alpha}{p}$ becomes insignificant while σ(p) continues to increase. Thus, another research opportunity involves finding algorithms, programming languages, and parallel architecture that, when used as a system, yield small σ's.

An important secondary question is how to determine that a good σ(p) has been found. Note that

$$\frac{1}{\bar{S}_p} = \frac{1}{p}\,[p(1-\alpha)+\alpha+p\sigma(p)] \cong \frac{1}{p}$$

if α is large and σ(p) is small. $\bar{S}_p$ is proportional to execution time in parallel mode, so a plot of execution time versus $\frac{1}{p}$ will reveal when high α and low σ have been achieved. Researchers at the US Los Alamos National Laboratory are working on parallel processing of several generic classes of scientific computation. One of these is particle-in-cell simulations that are widely used in plasma modeling. Figure 3 shows the results. Similar results

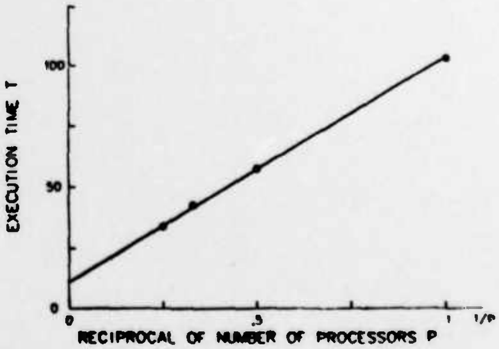


Figure 3. Execution time of particle-in-cell code on one to four processors of Univac 1100/84 as a function of the number of working processors.

have been obtained for fluid flow models and Monte Carlo simulations for up to $p = 8$.

Systems of a few tightly coupled high-performance processors have the potential to provide significant increases in computational capability. Realizing this potential will require development of highly parallel algorithms. These must be combined with suitable programming languages and architectures so that the overall implementation introduces little additional work relative to uniprocessor implementation. Experimentally validated models of performance will facilitate this research. In general, availability of experimental equipment will be a pacing factor in research on asynchronous systems.

The Ninth World Conference of IFIP was highly successful from the point of view of total attendance, number of nationalities present, and the high quality of papers presented. IFIP '86, the Tenth World Computer Congress, will be held in Dublin, Ireland, 1 through 15 September 1986.

Reference

Ware, W., "The Ultimate Computer," *IEEE Spectrum* (March 1973), 89-91.

DAT

FILM

UNITED STATES OF AMERICA

Computer logic

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Highlights of the conference included discussions of the design of Ada, the VLSI/ULSI chip, logic programming, multiprocessor data bases, high-performance computers, fifth-generation computers, dataflow machines, and high-speed processors.

ULTRALARGE. SCALE INTEGRATION

VERY LARGE SCALE INTEGRATION

size the need for training young workers in electronics to better prepare them for the opportunities that will exist in a society with increasing use of information systems and robots. Finally, he mentioned the interest of the government

dently developed packages for a given application has little control over the choices of names made by the producers of these packages. This fact has affected the naming rules of the language.